Computers Math. Applic. Vol. 25, No. 3, pp. 13–25, 1993
Printed in Great Britain.

0097-4943/93 \$5.00 + 0.00

THEORY AND ALGORITHMS FOR ANALYSING THE CONSISTENT REGION IN PROBABILISTIC LOGIC

D. BOUCHAFFRA
Centre de Recherche en Informatique appliquée aux Sciences Sociales
B.P. 47 X, 38040 Grenoble Cedex, France

(Received May 1991)

Abstract—This paper explores some topological features in order to analyse the consistent region in Probabilistic Logic. Using the L1 norm enables us to reduce and stabilize the consistent area associated with the probability of a predicate in a set of beliefs. The concept of facts and rules is approached as a particular problem. We present the program of the method used and propose an application to predicates in first-order logic. A study of the accuracy and the program complexity is made and compared to other methods.

1. INTRODUCTION

Dealing with uncertain information is a very common task in several artificial intelligence applications. The probabilistic logic seems to be quite necessary for manipulating uncertain data and rules [1]. One of the early expert systems in AI, which used a technique designed to handle uncertain knowledge, was MYCIN [2].

Other systems, for instance PROSPECTOR [3], used a Bayesian method to solve many problems. A lot of solutions with different interpretations are often encountered in the same problem. Thus, to obtain a unique solution, researchers began to investigate some heuristic methods based on finding the maximum-entropy probability distribution [4].

In this paper, we present a method which enables us to determine and reduce the consistent region associated with a set of beliefs. This paper is organized as follows. Section 2 introduces theoretical aspects like the construction of a probability space and a definition of a metric. In Section 3, we introduce the algebraic and geometric interpretation. In Sections 4 and 5, we apply the infinite and the L1 norm to first order logic predicates. Section 6 describes the practical aspect of the L1 method and its exploitation to the facts and rules concept. We present, in Section 7, the programming relative to the L1 method and the results obtained. Sections 8, 9 and 10 contain the complexity of the program, conclusion and plans for future work.

2. THEORETICAL ASPECT

2.1. The Definition of a Probability Space and a Metric

In any logical order, a predicate can be either true or false, and two sets of possible worlds are associated with this predicate. The first set, w_1, contains the worlds where the value of the predicate is true, and the other, w_2, contains the worlds where this value is false. Any configuration relative to this predicate must be inside one of these two sets. The stochastic idea is provided by imagining that any configuration belongs, respectively, to w_1 and w_2, with probability p_1 and $(1 - p_1)$. We can define a probability distribution over the sets of possible worlds associated with the different sets of possible truth values of the predicates [1]. The axioms of a probability measure are all respected: the exclusivity and exhaustivity of the possible worlds.

If S is any predicate in a set of beliefs, the probability of S can be taken as $P(S) = \sum P(w_i) * \Psi_{w_i}(S)$, where $P(w_i)$ is the probability that the actual world w_a (configuration) is equal to w_i,

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX the characteristic function Ψ is defined as

$$\Psi_{w_i}(S) = \begin{cases} 1, & \text{if } S \text{ is true in } w_i, \\ 0, & \text{if } S \text{ is false in } w_i, \end{cases}$$

where i is the index of possible worlds.

One can define a metric in the Euclidean space $\mathbb{R}^n$, where n is the number of possible worlds. This definition of both a probability and a metric associated with possible worlds gives a geometric and algebraic interpretation in the space of predicates.

3. THE ALGEBRAIC AND GEOMETRIC INTERPRETATION

Let $\mathcal{B}$ be the set $\{(\exists y)\,P(y),\ (\forall x)\,[P(x) \supset Q(x)]\}$, and let S be the predicate $(\exists z)\,Q(z)$. We are given probabilities for the predicates in $\mathcal{B}$ and want to compute bounds and analyze the probability associated with $(\exists z)\,Q(z)$.

We first have to compute the consistent sets of truth values for the predicates by the semantic-tree method, as illustrated in Figure 1. We represent predicates and their negations in skolem form, I, J and K [5]. The paths corresponding to inconsistent sets of truth values are closed by a black circle.

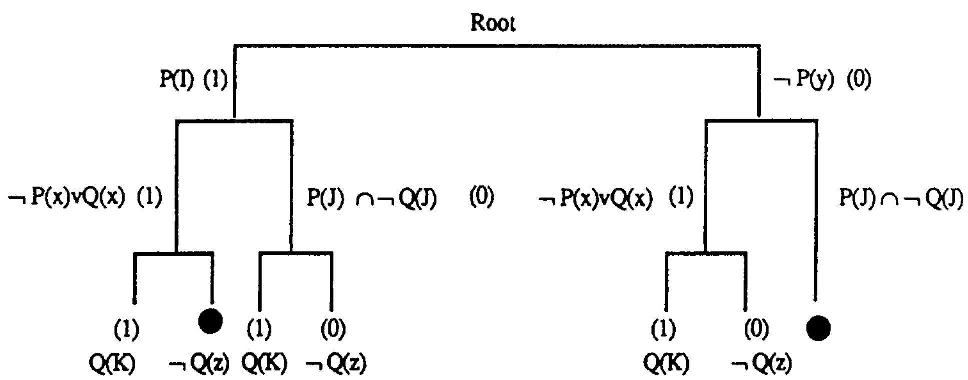

Figure 1. The binary tree associated with the set of predicates in first-order logic.

The consistent matrix extracted from the binary tree can be written

$$C = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

4. METHOD BASED ON THE MAXIMUM NORM

4.1. The Notion of Extreme Vector

We express, in the space of possible worlds, the extreme vector concept by the fact that the norm of any vector w is equal to 1; it can be written

$$\|w\|_\infty = 1,$$

which is equivalent to

$$\max\left|P(w_i)\right| = 1, \qquad \forall i \in \{1,2,3,4,5\}.$$

However, since $\sum P(w_i) = 1$, we know that if one component of the vector w is equal to 1, the others must be 0. Finally, we have obtained 5 extreme vectors (according to $\|\cdot\|_\infty$) corresponding to the base of $\mathbb{R}^5$ (the classical Euclidian space).

PROPOSITION 4.1.1. *The images of extreme vectors by a homomorphism are also extreme, according to the defined norm.*

PROOF. See [6]. ∎

Using this proposition, one can deduce the extreme vectors in the space of predicates. They are:

$$\Pi_1 = \begin{bmatrix}1\\1\\1\end{bmatrix}, \quad \Pi_2 = \begin{bmatrix}1\\0\\1\end{bmatrix}, \quad \Pi_3 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \Pi_4 = \begin{bmatrix}0\\1\\1\end{bmatrix}, \quad \Pi_5 = \begin{bmatrix}0\\1\\0\end{bmatrix}.$$

The maximal consistent region can, therefore, be plotted in Figure 2.

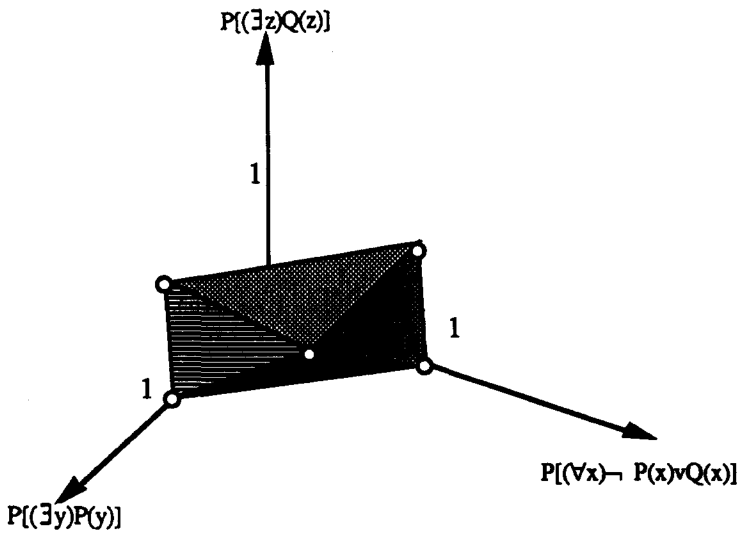

Figure 2. The maximal L_∞ consistent region.

The bounds relative to the predicate $(\exists z)\, Q(z)$ are found using geometrical properties as the mixed product

$$P\,[(\exists y)\, P(y)] + P\,[(\forall x)[P(x) \supset Q(x)]] - 1 \leq P\,[(\exists z)\, Q(z)] \leq 1. \tag{1}$$

4.2. The Multitudinous Mathematical Solutions

Many different values are associated with the probability of the predicate $(\exists z)\, Q(z)$ in the inequation (1). These multitudinous mathematical solutions often have different interpretations, especially when the difference between the values is large. This is a problem in many different fields like expert systems or other processes based on probabilistic inference [3]. When using the L1 norm, it is possible to reduce the set of solutions. We shall see in the next section that the manner of reducing this set is very important. The reduction is made uniformly until we obtain a fixed set containing only neighbouring solutions. Among these neighbouring solutions the choice of an "ideal" one is much easier. We have seen in [7] that a unique solution is obtained only if the consistent matrix is an isometry. Preserving both the probabilistic approach and obtaining a unique solution seems to be incompatible but reducing uniformly the set of solutions is quite possible, as we discover in the next section.

5. METHOD BASED ON THE L1 NORM AND THE NOTION OF EXTREME VECTOR

The notion of extreme vector is, according to the L1 metric, that the norm of any possible world vector $w = (w_i)$ deduced from the example can be written

$$||w||_1 = \sum_{i=1}^{i=5} P(w_i).$$

PROPOSITION 5.1. *Any vector defined in the space of possible worlds is extreme according to this norm.*

PROOF. This is due to the fact that any probability vector must satisfy $\sum P(w_i) = 1$. ∎

Before giving the method and a solution, we present some topological properties. Let $\mathcal{P} = \{W = [P(w_1), P(w_2), \dots, P(w_n)]^{\mathrm{T}} / \sum P(w_i) = 1 \text{ and } \forall i, P(w_i) = Pi \geq 0\}$.

LEMMA 5.2. *$\mathcal{P}$ is the space of possible worlds and it is a convex and compact set in the Euclidean space $\mathbb{R}^n$.*

PROOF. $\mathcal{P}$ is convex and closed, because it is the intersection of closed convex sets; it is also bounded, because it is contained inside the unity sphere $||w||_1 = 1$. ∎

THEOREM 5.3. *Let U be a linear operator in $\mathbb{R}^n$ and $M = (M_{ij})$ its associated matrix relative to the classical base of $\mathbb{R}^n$. If $\sum M_{ij} = n, \forall j$ [$(n-1)$ is the number of predicates in the set of beliefs], $M_{ij} \geq 0$, then we can write the following assertions:*

(1) $U(\mathcal{P}) \subset n\mathcal{P}$.
(2) $||\Pi||_1 = n\,||w||_1$.

PROOF. When using the hypothesis of this theorem, one can write:

$$||M * w||_1 = \sum_{i=1}^{i=n}\sum_{j=1}^{j=n} M_{ij}\, w_j = \sum_{j=1}^{j=n} w_j \sum_{i=1}^{i=n} M_{ij} = n \sum_{j=1}^{j=n} w_j,$$

and both of the relations 1 and 2 are proved. ∎

PROPOSITION 5.4. *The sets $\mathcal{P}$, $U(\mathcal{P})/n$, $U^2(\mathcal{P})/n^2, \dots, U^k(\mathcal{P})/n^k, \dots$, form a decreasing sequence with an intersection D different from $\emptyset$.*

PROOF. $U^k(\mathcal{P})/n^k$ is a decreasing sequence when applying the previous theorem, so any finite intersection of the sequence is different from $\emptyset$. Now, for all k, U_k is continuous, because it is a linear operator in a finite dimension space, thus, $U^k(\mathcal{P})/n^k$ is compact, and $\cap U^k(\mathcal{P})/n^k = D \neq \emptyset$. ∎

PROPOSITION 5.5. *$U(D) = D$ and, for each probability vector $V \in D$, $U(V) \in D$.*

PROOF. If $V \in D$, for all $k \geq 0$, $V \in U^k(\mathcal{P})/n^k$, therefore, $U(V) \in U^{k+1}(\mathcal{P})/n^k$, e.g., $U(V) \in D$. If $W \in D$, by the definition of D, for all $k \geq 0$, $\exists V_k \in U^k(\mathcal{P})/n^k$ such that $W = U(V_k)$. As $\mathcal{P}$ is compact, the sequence V_k has at least one adherence value V, then $V \in D$, since $V_k \in \cap U^i(\mathcal{P})/n^i$ with $i \leq k$, and the $U^i(\mathcal{P})/n^i$ are closed and decreasing. In conclusion, as $U(D) \subset D$ and $D \subset U(D)$, then $U(D) = D$. ∎

THEOREM 5.6. *If $M(k) = (M_{ij}(k))$ is the matrix associated with U^k/n^k, relative to the canonical base of $\mathbb{R}^n$, then a strictly increasing sequence, $k_1, k_2, k_3, \dots$, of natural numbers exists so that, for all i and j, the sequence $(M_{ij}(k_p))$ tends to the limit matrix $\beta = (\beta_{ij})$, when p tends to infinity and $L(\mathcal{P}) = D$, where L is a linear operator associated with the matrix $\beta = (\beta_{ij})$.*

PROOF. For all (i,j,k), we can write: $0 \leq M_{ij}{}^k \leq 1/n^k$; so, for all (i,j), the sequence $(M_{ij}{}^k)$ possesses an adherence value (Bolzano-Weiertrass). So, by setting in order the finite set of indices i and j, we can, by n_2 extractions of the sequence, find an increasing sequence of natural numbers (k_p) so that, for all i and j, the sequence $(M_{ij}(k_p))$ tends to $\beta = (\beta_{ij})$ (adherence value) when p tends to infinity. For all $p \geq 1$, we can write, using the definition of L : $L(\mathcal{P}) \subset U^{k_p}(\mathcal{P})/n^{k_p}$, thus, also $L(\mathcal{P}) \subset D$ because $U^i(\mathcal{P})/n^i$ is a decreasing sequence. Let $V \in D$; for any natural number p, we can find $V_{k_p} \in \mathcal{P}$ such that $V = U^{k_p}(V_{k_p})/n^{k_p}$ [6]. If W is an adherence value associated with the sequence V_{k_p}, we obtain the result by using, for example, the following inequality

$$||U^{k_p}(V_{k_p})/n^{k_p} - L(W)|| \leq ||U^{k_p}(V_{k_p})/n^{k_p} - L(V_{k_p})|| + ||L(V_{k_p}) - L(W)||.$$ ∎

PROPOSITION 5.7. *$\mathcal{P}$ is the convex hull relative to $\{e_i\}$, D is the convex hull relative to $\{L(e_i)\}$, where $\{e_i\}$ is the canonical base associated with $\mathbb{R}^n$.*

PROOF. This is due to the fact that L is a linear operator and $L(\mathcal{P}) = D$. ∎

6. PRACTICAL ASPECT

In this section, we emphasize the solutions provided by the use of this method. The analysis of the theoretical results and the determination of the solutions are presented. This method is also applied to the problem of facts and rules, considered with their probabilities.

6.1. The Analysis of the Theoretical Results

The original problem is $C\,w = \Pi$, where C is the consistent matrix, w is the probability vector associated with the possible worlds, Π is the vector probability relative to the predicates. Some methods using entropy of the distribution w have been used, but they remain approximative and subjective. The consistent matrix is

$$C = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

We complete the matrix C by adding other rows, so that the sum of the elements of each column will be equal to the number of possible worlds. The new matrix C_a is square and the value of each element is positive.

C_a corresponds to the linear operator U used in Theorem 5.3., and it is as follows:

$$C_a = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \\ \boxed{\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 2 & 3 \end{matrix}}$$

The problem becomes $C_a * w = \Pi_a = [P[(\exists y)P(y)], P[(\forall x)(P(x) \supset Q(x)], P[(\exists z)\,Q(z)], \mathcal{A}_1, \mathcal{A}_2]^{\mathsf{T}}$, where $\mathcal{A}_i$ are the components deduced from the two added rows of the matrix C_a. In order to find $\beta = (\beta_{ij})$, we have to multiply the square new matrix $C_a * 1/n = {C_a}^d$ (d means division by n) by itself p times; the number p of iterations used is the reduction factor of the method. In conclusion, we obtain the following diagram

$$C \to C_a \to {C_a}^d \to \left({C_a}^d\right)^2 \to \left({C_a}^d\right)^3 \to \cdots \to \left({C_a}^d\right)^p.$$

$({C_a}^d)^p$ corresponds to the limit matrix $\beta = (\beta_{ij})$ of the topological results; this matrix is associated with the linear operator L. As the approximated limit matrix $({C_a}^d)^p$ is found, then D is obtained by the convex hull $({C_a}^d)^p(e_i)$, where $\{ei\}$ is the canonical base of the vector space.

6.2. The Determination of the Solutions

The main results found when using those topological properties are:

(1) $L(\mathcal{P}) = D$, D is the convex hull associated with $\{L(e_i)\}$ and L is the limit operator,
(2) $D \subset U(\mathcal{P})/n$, and
(3) $U(D) = D$, D is invariant by the linear operator U.

Using the relation $D \subset U(\mathcal{P})/n$, one can write

$$\forall w = \{P(w_i)\} = [p_1, p_2, p_3, \dots, p_n]^{\mathsf{T}} \in D \Rightarrow w \in \frac{U(\mathcal{P})}{n},$$

which is equivalent to

$$\exists \Pi = [\pi_1, \pi_2, \pi_3, \dots, \pi_n]^{\mathsf{T}} \in U(\mathcal{P}) \quad \text{such that } \Pi = n\,w, \tag{2}$$

(n is the number of possible worlds or the dimension of the vector space).

REMARK 6.2.1. The Assertion 2 is the characterization of a reduced consistent region. This result enables the expert to choose solutions among a reduced set of predicates vectors. It is easy for the expert to choose a solution among a small number of neighbouring solutions. However, in [7], we showed that the unique solution for Π is obtained only if the operator U is an isometry, so the process will stabilize after a certain number of iterations. This remark will oblige searchers to try other logics which are not necessarily inherent to a probability measure [8].

PROPOSITION 6.2.2. *Any point W considered inside D can be written as a convex combination associated with $L(e_i)$,*

$$W = \sum_{i=1}^{i=n} \lambda_i L(e_i) \quad \textit{subject to} \sum_{i=1}^{i=n} \lambda_i = 1 \textit{ and } \forall i,\ \lambda_i \in [0,1].$$

PROOF. Because $\{L(e_i)\}$ is the set of vectors which generates the convex hull D. ∎

PROPOSITION 6.2.3. *If one increases the number of multiplications of the matrix $({C_a}^d)$ by itself (the reduction factor of the method), one tends to approach D and, therefore, the image Π of any vector in D also approaches D.*

PROOF. Because $U(D) = D$ and D is obtained by the limit operator L. ∎

6.3. Generalization to Other Topologically Equivalent Norms

DEFINITION 6.3.1. *Two norms N_1 and N_2 defined in a finite dimension space $\mathcal{E}$, are said to be topologically equivalent if any open set for N_1 is also an open set for N_2, and reciprocally [6]. This can be written as*

$$\exists\, k > 0, \textit{ such that: } N_1(x) \le k\, N_2(x) \textit{ and } N_2(x) \le k\, N_1(x), \quad \forall x \in \mathcal{E}.$$

REMARK 6.3.2. The results obtained when using the L1 norm can be preserved when dealing with other topologically equivalent norms like the infinite norm or the L2 norm [9].

The assertion "The sets $\mathcal{P}$, $U(\mathcal{P})/n$, $U^2(\mathcal{P})/n^2, \dots, U^k(P)/n^k, \dots$, form a decreasing sequence with an intersection D different from $\emptyset$" is independent of any metric.

6.4. Facts and Rules

This method can be used for a set of beliefs containing facts and rules. If we associate the probability P_1 and P_2, respectively, with the rule R_1 and the fact F_1, we can compute the probability associated with F_2 where $F_2 \Leftarrow R_1 \cap F_1$. This probability can be written

$$P(F_2) = f[P(R_1), P(F_1)],$$

where f is a function associated with the surface of the convex hull and can be computed. This method can be generalized to a certain number of facts and rules contained in the knowledge base. Let

$$\mathcal{B} = \left\{ \frac{F_1}{p_1}; \frac{F_2}{p_2}; \dots; \frac{F_n}{p_n}; \frac{R_1}{p_{n+1}}; \dots; \frac{R_m}{p_{n+m}}; \frac{F_{n+1}}{p_{n+m+1}}; \frac{F_{n+2}}{p_{(n+m+2)}} \right\}$$

be the set of beliefs. We suppose that the probabilities of F_i and R_j are known $(1 \le i \le n)$, $(1 \le j \le m)$, and want to calculate the probabilities associated with the goals F_{n+1} and F_{n+2}. This procedure begins by computing the consistent matrix associated with the set

$$\mathcal{B}_0 = \left\{ \frac{F_1}{p_1}; \frac{F_2}{p_2}; \dots; \frac{F_{n+1}}{p_{n+m+1}} \right\}.$$

This method can be applied to determine the consistent region associated with $\mathcal{B}_0$ and, by an induction process, one can compute the consistent matrix associated with

$$\mathcal{B} = \mathcal{B}_0 \cup \left\{ \frac{F_{n+2}}{p_{(n+m+2}} \right\}.$$

The probability values associated with F_{n+1} and F_{n+2} are not unique. Therefore, in order to reduce the number of solutions, the L1 method must be employed with a large number of iterations.

7. THE PROGRAMMING OF THE L1 METHOD

The programming of this method is as easy to define as any iterative process [10]. We choose e PASCAL language in the Suntools Environment to treat the case of the example.

```
OGRAM L1ConsistentArea (input,output);
PUT: A consistent matrix whose elements are the truth values associated with
edicates, the number of iterations, or the reduction factor, and the convex
rameters λ1,λ2,λ3,λ4(∑λi = 1 and λi ≥ 0).
TPUT: The matrix power, the number of iterations, this matrix enables us to
ot the convex hull D, and the vector of predicates Π = [π1,π2,π3,π4,π5] associated
th the fixed point W ∈ D.

CONST MAX = 100;
TYPE matrix = array [1..Max,1..Max] of real;
     vector = array [1..Max] of real;
VAR  p,Worlds, Predicates, Iterations  :  integer;
     C, Ca, Cad, Power, product        :  matrix;
     Lambda, WorldVector, Pi           :  vector;

PROCEDURE ReadingConsistentMatrix;
INPUT: The number of predicates, the number of possible worlds, the consistent
matrix elements and the number of iterations.
OUTPUT: The elements indexes of the consistent matrix.
VAR I,J : integer;
BEGIN
  Write('Please, enter the number of predicates contained in the set of beliefs '
  Readln(Predicates);
  Write('the number of possible worlds =  ');
  Readln(Worlds);
  Writeln('Would you please, enter the consistent matrix : ');
  FOR  I:=1 TO Predicates DO
  FOR  J:=1 TO Worlds DO
  BEGIN
    Write('  C[',I:2,',',J:2,'] = ');
    Readln(C[I,J]);
  END;
  Writeln;
  Write('The number of iterations corresponding to the reduction factor
  is =  ');
  Readln(p);
  Iteration:=p;
  Writeln;
END;  {ReadingConsistentMatrix}

PROCEDURE PrintConsistentMatrix;
OUTPUT: Print the elements of the matrix A.
VAR I,J : integer;
BEGIN
  Writeln('C = ');
  FOR I:=1 TO Predicates DO
  BEGIN
    FOR J:=1 TO Worlds DO
    Write(C[I,J]:10:3);
    Writeln;
  END;
END; {PrintConsistentMatrix}

PROCEDURE SquareMatrix(VAR A,B : matrix);
INPUT: The matrix A corresponding to the consistent matrix.
OUTPUT: The matrix B which contains A, where some rows are added to A
such that the sum of the elements of each column is equal to the number
of possible worlds.
VAR  I, J, K   : integer;
     R, D, Sum : real;
BEGIN
  R:= Worlds - Predicates;
```

```
    FOR J:=1 TO Worlds DO
      BEGIN
      Sum:= 0.0;
      K:= Predicates + 1;
      FOR I:= 1 TO Predicates DO
        Sum:= Sum + A[I,J];
        D:= Worlds - Sum;
        WHILE  K < Worlds DO
          BEGIN
            A[K,J]:= 1;
            K:=K + 1;
          END;
        A[K,J]:= (D - R + 1);
      END;
    FOR J:= 1 TO Worlds DO      {Storage of the matrix A in B}
    FOR I:= 1 TO Worlds DO
    B[I,J]:= A[I,J];
  END;   {SquareMatrix}

  PROCEDURE PrintSquareMatrix;
  OUTPUT: Print the elements of the matrix Ca.
  VAR I, J  : integer;
  BEGIN
    Writeln('Ca = ');
    Writeln;
    FOR I:= 1 TO Worlds DO
    BEGIN
      FOR J:= 1 TO Worlds DO
      Write(Ca[I,J]:10:3);
      Writeln;
    END;
    Writeln;
    Writeln;
  END; {PrintSquareMatrix}

  PROCEDURE Division(A :  matrix ; VAR B :  matrix);
  OUTPUT: The matrix B whose elements correspond to the elements of A divided
  by the number of possible worlds.
  VAR I,J : integer;
  BEGIN
    FOR I:=1 TO Worlds DO
    FOR J:=1 TO Worlds DO
    B[I,J]:= A[I,J]/Worlds;
  END;  {Division}

  PROCEDURE Heading ;
  OUTPUT: Message relative to the computation of the matrix Cad.
  BEGIN
    Writeln('                            ',p:8');
    Writeln('Computation of the matrix Cad, where p is the number of
      iterations.');
  END;  {Heading}
```

In order to obtain a fast algorithm, the multiplication of the matrix by itself is processed in the following manner. If the binary representation of p is equal to $b_k, \ldots, b_0$, this means that: $p = 2^k\, b_k + 2^{(k-1)}\, b_{(k-1)} + \cdots + 2\, b_1 + b_0$, thus one can write

$$\left(C_a{}^d\right)^p = \left(C_a{}^d\right)^{2^k\, b_k} * \cdots * \left(C_a{}^d\right)^{2b_1}.$$

One can, therefore, compute $(C_a{}^d)^p = P_k$ with $P_0 = P_{i+1}$ identity, $(C_a{}^d)_0 = (C_a{}^d)$ and $(C_a{}^d)_{i+1} = (C_a{}^d)_i^2$, and if $b_i = 1$, then $P_{i+1} = P_i * (C_a{}^d)_i$, else $P_i + 1 = P_i$.

The integers n and p are, respectively, the dimension of the vector space and the power of the matrix $(C_a{}^d)$.

```
PROCEDURE InitializeIdentity(VAR Id : matrix);
OUTPUT: Initialize the matrix power to identity.
VAR I, J :  integer;
BEGIN
  FOR I:=1 TO Worlds DO
  BEGIN
    FOR J:=1 TO Worlds DO
    Id[I,J]:=0.0;
    Id[I,I]:=1.0;
  END;
END;  {InitializeIdentity}

PROCEDURE PowerIteration(VAR A , B : matrix);
OUTPUT: The matrix B which corresponds to raising the matrix A to the power p.
VAR I, J, K  :  integer;
    T        :  matrix;
BEGIN
  WHILE p>0 DO
  IF odd(p) THEN
  BEGIN
    p:= p - 1;   {B= B * A}
    FOR I:=1 TO Worlds DO
    BEGIN
      FOR J:=1 TO Worlds DO
      Li [J]:= B[I,J];
      FOR J:=1 TO Worlds DO
      BEGIN
        B[I,J]:=0.0;
        FOR K:=1 TO Worlds DO
        B[I,J]:= B[I,J] + (Li[K] * A[K,J]);
      END;
    END;
  END;
ELSE
BEGIN
  p:= p Div 2;    {A= A * A  and  T=A}
  FOR I:= 1 TO Worlds DO
  FOR J:= 1 TO Worlds DO
  T[I,J]:= A[I,J];    T= A
    FOR I:= 1 TO Worlds DO
    FOR J:= 1 TO Worlds DO
    BEGIN
      A[I,J]:=0.0;
      FOR K:= 1 TO Worlds DO
      A[I,J]:= A[I,J] + (T[I,K] * T[K,J]);  {A=T * T}
    END;
  END;
END;  {PowerIteration}

PROCEDURE PrintPowerIteration (R : matrix);
OUTPUT: Print the matrix R.
VAR  I, J :  integer;
BEGIN
  Writeln;
  Writeln;
  Writeln(Iteration:3);
  Writeln('Cad  = ');
  Writeln;
  FOR I:=1 TO Worlds DO
  BEGIN
    FOR J:=1 TO Worlds DO
    Write(R[I,J]:10:3);
```

```
    Writeln;
  END;
  Writeln;
  Writeln;
END;  {PrintPowerIteration}

PROCEDURE StorageConvexParameters(VAR B : vector);
INPUT: The convex parameters whose sum is equal to 1.
OUTPUT: The vector B containing the convex parameters.
VAR I : integer;
    S : real;
BEGIN
  Writeln('Will you introduce',(Worlds - 1):2,'positive convex parameters,');
  Writeln('these parameters must belong to the interval [0..1].');
  Writeln('The last parameters is deduced by the fact that their sum is
  equal 1.');
  Writeln;
  S:= 0.0;
  FOR I:= 1 TO Worlds - 1 DO
  BEGIN
    Write('The convex parameter',i:2,' is equal to: ');
    Readln(B[I]);
    S:= S + B[I];
  END;
  B[Worlds]:= 1 - S;
  Writeln('the convex parameter',Worlds:3,' is equal to: ',1-S:5:3);
  Writeln('the sum of convex parameters is equal to ',(B[Worlds] + S):5:3);
  Writeln;
  Writeln;
END; {StorageConvexParameters}

PROCEDURE MatrixScalarProduct(VAR T : matrix ; L : matrix ; B : vector);
INPUT:  The limit matrix L and the vector containing the convex parameters.
OUTPUT: The matrix T whose columns are multiplied by the convex parameters.
VAR I,J : integer;
BEGIN
  FOR J:=1 TO Worlds DO
  FOR I:=1 TO Worlds DO
  T[I,J]:= B[J] * L[I,J];
  Writeln;
END; {MatrixScalarProduct}

PROCEDURE ConvexCombination(T: matrix ; VAR W : vector);
INPUT:  The matrix T corresponding to the limit operator L.
OUTPUT:  The vector W, which is equal to the convex combination of the column
associated with the matrix T.
VAR I,J : integer;
    S : real;
BEGIN
  S:= T[1,1];
  FOR I:=1 TO Worlds DO
  BEGIN
    FOR J:= 1 TO Worlds - 1 DO
    S:= S + T[I,J+1];
    W[I]:= S;
    Write(W[I]:10:3);
  END;
  Writeln;
  Writeln;
  Writeln('the previous vector is the possible world vector.');
  Writeln;
END;   {ConvexCombination}
```

```
PROCEDURE EvaluatePredicate(A : vector);
INPUT: The vector A corresponding to a possible world which is inside
the convex hull D.
OUTPUT: The image vector predicate associated with the vector A.
VAR I : integer;
BEGIN
  Writeln('The following vector is the predicate vector.');
  Writeln;
  FOR I:=1 TO Worlds DO
  BEGIN
    PI[I]:= Worlds * A[I];
    Write(Pi[I]:10:3);
  END;
  Write('.');
END;  {EvaluatePredicate}

{Main: L1consistentArea}
BEGIN
  ReadingConsistentMatrix;
  PrintConsistentMatrix;
  SquareMatrix(C,Ca);
  PrintSquareMatrix;
  Division(Ca,Cad);
  Heading;
  InitializeIdentity(Power);
  PowerIteration(Cad,Power);
  PrintPowerIteration(Power);
  StorageConvexParameters(Lambda);
  MatrixScalarProduct(Product,Power,Lambda);
  ConvexCombination(Product,WorldVector);
  EvaluatePredicate(WorldVector);
END. {L1ConsistentArea}
```

8. THE PROGRAM COMPLEXITY

The study of the program complexity focuses essentially on the analysis of the raising the matrix $(C_a{}^d)$ to a certain power. This power is called the reduction factor, because it reduces the consistent region. The naive algorithm which consists of multiplying the matrix $(C_a{}^d)\, p$ times leads to p matrix multiplications. If processed in this way, one obtains a program whose complexity is equal to $O(p\,n^3)$. The program used in this paper is much more efficient because the number of matrix multiplications is $\log_2 p$ $(p > 1)$ and, therefore, the complexity becomes equal to $O(n^3 \log_2 p)$.

However, the complexity associated with the intermediate procedures is insignificant. Other approximative methods using entropy of the probability distribution associated with the possible world vector have been tried. Unfortunately, when the consistent matrix is large, those methods become impractical.

In the next section, we enclose the results obtained from the example. The predicate vector obtained is associated with the possible world vector. This last vector is the convex combination of the limit matrix $(C_a{}^d)^p$ columns.

For $p = 100$, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0 \Rightarrow \lambda_5 = 1$, one can read the probability vector associated with the predicates by considering only the first three dimensions.

9. OUTPUTS

```
Please, enter the number of predicates contained in the set of beliefs: 3
The number of possible worlds = 5
```

```
Would you please, enter the consistent matrix:

    C [ 1, 1 ] = 1
    C [ 1, 2 ] = 1
    C [ 1, 3 ] = 1
    C [ 1, 4 ] = 0
    C [ 1, 5 ] = 0
    C [ 2, 1 ] = 1
    C [ 2, 2 ] = 0
    C [ 2, 3 ] = 0
    C [ 2, 4 ] = 1
    C [ 2, 5 ] = 1
    C [ 3, 1 ] = 1
    C [ 3, 2 ] = 1
    C [ 3, 3 ] = 0
    C [ 3, 4 ] = 1
    C [ 3, 5 ] = 0

The number of iterations corresponding to the reduction factor is = 100

C =
    1.000     1.000     1.000     0.000     0.000
    1.000     0.000     0.000     1.000     1.000
    1.000     1.000     0.000     1.000     0.000

C =
    1.000     1.000     1.000     0.000     0.000
    1.000     0.000     0.000     1.000     1.000
    1.000     1.000     0.000     1.000     0.000
    1.000     1.000     1.000     1.000     1.000
    1.000     2.000     3.000     2.000     3.000

The following matrix is called Cad.

    0.2000000     0.2000000     0.2000000     0.0000000     0.0000000
    0.2000000     0.0000000     0.0000000     0.2000000     0.2000000
    0.2000000     0.2000000     0.0000000     0.2000000     0.0000000
    0.2000000     0.2000000     0.2000000     0.2000000     0.2000000
    0.2000000     0.4000000     0.6000000     0.4000000     0.6000000
```

Computation of the matrix $(Cad)^{100}$, where 100 is the number of iterations

Cad^{100} =

```
    0.0588235     0.0588235     0.0588235     0.0588235     0.0588235
    0.1529412     0.1529412     0.1529412     0.1529412     0.1529412
    0.0823529     0.0823529     0.0823529     0.0823529     0.0823529
    0.2000000     0.2000000     0.2000000     0.2000000     0.2000000
    0.5058824     0.5058824     0.5058824     0.5058824     0.5058824

Will you introduce 4 positive convex parameters,
These parameters must belong to the interval [0..1].
The last parameter is deduced by the fact that their sum is = 1.

The convex parameter 1 is equal to: 0
the convex parameter 2 is equal to: 0
the convex parameter 3 is equal to: 0
the convex parameter 4 is equal to: 0
the convex parameter 5 is equal to: 1.000000000

0.05882    0.15294     0.08235     0.20000     0.50588

The previous vector is the possible world vector.
The following vector is the predicate vector associated with the previous
possible world vector.

0.29412     0.76471     0.41176     1.00000     2.52941.
```

10. CONCLUSION

In this paper, we have presented a general method to determine the probability associated with some added predicates in a set of beliefs. We notice that using the probabilistic approach, one can reduce the consistent region associated with the predicates. However, the solution associated with the predicate probability is not unique. We have seen that the L1 method is general and can be applied to facts and rules. The set of solutions is stabilized after a certain number of iterations and the solution is unique only if the consistent matrix is an isometry. This last condition corresponds to the case where the space of predicates is a probability space [7].

Obtaining a unique solution in probabilistic logic seems to be incompatible with the classical binary logic. Theorem 5.3. shows that a unique solution is obtained only if the sum of each column of the matrix $M = (M_{ij})$ is equal to 1. This seems to be inappropriate when using a binary representation.

Research must still be done in order to complete the model by adding "some physical constraints" which are absent in the probabilistic approach. These constraints can reduce the conflict among all mathematical solutions.

REFERENCES

1. N.J. Nilsson, Probabilistic logic, *Artificial Intelligence* **28** (1), 71–87 (1986).
2. E.H. Shortiffe, *Computer-Based Medical Consultations: MYCIN*, Elsevier, New York, NY, (1976).
3. R.O. Duda, P. Hart and N.J. Nilsson, Subjective Bayesian methods for rule-based inference systems, Techn. Note 124, SRI International, AI Center, Menlo Park; also: *Proceedings of the NCC*, AFIPS Press, (1976).
4. P. Cheeseman, A method of computing generalized Bayesian probability values for expert systems, In *Proceedings Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Fed. Rep. Germany, William Kaufmann, Los Altos, CA, (1978).
5. S.C. Kleene, *Logique Mathématique*, Jacques Gabay, Paris, (1987).
6. J. Dieudonné, *Foundations of Modern Analysis*, Academic Press, New York, NY, (1960).
7. D. Bouchaffra, A relation between isometries and the relative consistency concept in probabilistic logic, In *Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics*, Dublin, pp. 22–26, (July 1991).
8. L.A. Zadeh, Fuzzy logic and approximate reasoning, *Synthese* **30**, 407–428 (1975).
9. D. Bouchaffra, Consistent regions in probabilistic logic when using different norms, *Proceedings of the Third Workshop on Artificial Intelligence and Statistics*, Miami, Fort Lauderdale, FL, pp. 5.1–5.5, (January 1991).
10. D.E. Knuth, *Fundamental Algorithms: The Art of Computer Programming*, Vol. 1, Second Edition, Addison-Wesley, Reading, MA, (1973).